

VIRTUAL NETWORK WITH ADAPTIVE DISPATCHER**CROSS-REFERENCE TO RELATED APPLICATIONS**

[01] This application relates to and claims priority from U.S. Provisional Application serial number 60/____,____ (Attorney Docket Number MS188903.1), filed October 16, 2001, and U.S. Provisional Application serial number 60/____,____ (Attorney Docket Number 188903.2), filed October 19, 2001, each of which is herein incorporated by reference.

FIELD OF THE INVENTION

[02] The invention relates generally to computer networks. More specifically, the invention relates to a virtual network that adaptively routes messages based on message contents.

BACKGROUND OF THE INVENTION

[03] Various levels of abstraction exist within computer architecture, from the physical representation of ones and zeros to high-level application programs. When computers were initially developed, a low-level programming language commonly referred to as machine language was generally used to control their operation. However, in order to create the same program for two different computer platforms with different machine languages, programmers had to write the program twice—once in each platform's machine language.

[04] Computer programmers learned that machine language could be abstracted by creating higher-level programming languages, such as C and Pascal, and then providing a compiler for each platform on which the program was to be used. When a program was written in one of these higher-level programming languages, the program could be compiled to run on each specific machine, without having to rewrite the source program for each machine. Abstractions in this regard continued, resulting in the more recent development of virtual machines.

[05] The notion of a virtual machine is well known in the art of computer science. A virtual machine is an intermediate representation that is not tied to the specific details of a particular computer hardware architecture. Typically a virtual machine will guarantee certain semantics that remain identical regardless of the hardware used to implement it. Therefore a program which has been written for such a machine can be executed on different hardware systems without modification. Thus, one advantage of a virtual machine is that its operational semantics remain constant from one computer program to the next regardless of the origin or operating requirements of any one computer program.

[06] Computer networks are dependent on the underlying physical hardware and network protocols on which the network is constructed. These protocols in turn are dependent on the underlying network architecture on which they are implemented. As a result, network applications must be rewritten for each network on which they are to be used. In addition, in order for two machines to communicate over a network, each machine must understand how to communicate over the specific network, i.e., each machine must have the appropriate network drivers to communicate.

[07] One level of abstraction that has been implemented in computer networks is the use of a TCP/IP protocol stack, as implemented according to the OSI seven-layer network model. TCP/IP abstracts some notions of network protocols, allowing two machines that each understand the TCP/IP protocols to effectively communicate with each other. However, even using TCP/IP, each machine must, at some level, be able to understand network routing and topology, bindings, and DNS resolution. That is, each computer on a network must still have substantial network support utilities installed in order to effectively communicate over the network, because the OSI model only virtualizes the physical wire between the machines, and not the network through which the machines communicate.

[08] For example, TCP/IP requires applications to understand the concepts of ports and IP addresses. Ports and IP addresses, in turn, require applications to understand DNS name resolution, network topology, transport bandwidths and end-to-end routing. Thus, while simplifying the model for exchanging ordered sequences of bytes in a reliable manner, the

application still must deal directly with many network level concepts and details. The OSI model does not address higher-level constructs, such as naming, routing, and quality of service, as needed by network applications.

[09] Another shortcoming of conventional networks is the inability to adapt and rehabilitate after a message error or network failure. Present networks cannot easily adapt automatically when machines are added, moved, or removed. That is, a user typically must edit routing tables to inform the network of the change.

[10] In addition, network failures are not easily fixed, other than by maintaining redundant machines that perform the same function. That is, if a first machine fails, then the second (backup) machine takes over the first machine's functions. However, if the second machine subsequently fails, and there is no third machine that performs the same functions, the network will suffer as a result. Known networks are not self-healing. Thus, an advanced network that overcomes these problems is needed.

[11] Another shortcoming of conventional networks is their inability to dynamically route network messages based on message contents. Known routers by Cisco Systems, Inc. are capable of routing messages based on predefined criteria, but are not dynamically programmable to support user-extensible routing behavior based on message content. This inability makes them inappropriate for systems in which applications can control transformations and processing of messages, in addition to the traditional routing and QoS requirements.

[12] It would be an advancement in the art to provide a method and system that solves some or all of the above-identified problems.

BRIEF SUMMARY OF THE INVENTION

[13] Aspects of the invention may be used to virtualize a network to allow programmers to write platform independent network computer applications. A virtual network may be created by adding an abstraction layer (software or hardware) between the applications providing network services and the underlying network of systems that executes that code,

for example, between layers 6 and 7 in the OSI network model. One or more virtualized components may be inserted, including synchronization, eventing, messaging, naming, groups, addressing, and routing components.

[14] By abstracting the networking system, the network may more efficiently and securely provide services inherently to the applications and services built on top of it. For example, services such as reliability, security, platform independence, scale-out, edge networking, and location independence may easily be provided. Also, the system may adapt to physical topology changes and automatically “heal” from failures. The virtual network is responsible for mapping code onto the physical topology of the network and transparently adapting that mapping. Additionally, developers may benefit from isolation of their services.

[15] By combining the virtual network with a virtual machine, a distributed, partitionable virtual network can be created where an application can be written once and run on any machine. That is, a network application written for use in a virtual network, and on a computer running a virtual machine, does not need to be rewritten because it is able to run on any machine that operates the same virtual machine and is connected to the virtual network.

[16] The virtual network also provides adaptive reconfiguration capabilities. Suppose that machine A sends a message to machine B over a network, and B replies back to machine A. However, before machine A can send a second message to machine B, machine B moves (e.g., to another IP address). According to one embodiment, a virtual network may resolve itself and adapt to the changed location such that the message is still delivered to B’s new location. The address change may take place transparently so that the applications running on the network(s) never know (or need to know) that a change in B’s location was made. No restrictions are placed on the set of locations to which each machine may be moved because abstraction is moved from the machine level to the network level. After a device has been moved, once it identifies itself to the network at the new location the virtual network has the ability to update itself so that the routing to the machine can continue to operate uninterrupted. This ability extends what is currently possible within single administrative

domains to multiple administrative domains, enabling location mobility to extend across organizations.

[17] A first embodiment of the invention provides an apparatus that includes a message dispatcher that routes and dispatches messages. Each message is routed based on an arbitrary portion of the message's contents. There is also an interface through which network application programs communicate with the message dispatcher to define the arbitrary portion of the message's contents on which the message is routed.

[18] In another embodiment of the invention, there is a data processing apparatus that includes a message dispatcher module, a transport adapter for interfacing the message dispatcher to a transport protocol, an interface through which application programs communicate with the message dispatcher, and stored rules instructing the message dispatcher to route a first network message based on a first attribute of said first network message, and route a second network message based on a second attribute, different from the first attribute, of the second network message. The first and second attributes are selected from a set of headers and data contained in each network message.

[19] Another embodiment provides a method for routing network messages. A message dispatcher routes a first network message based on a first attribute of the first network message. The message dispatcher routes a second network message based on a second attribute, different from the first attribute, of the second network message. The first and second attributes may be any field selected from a set of headers and data of each network message.

[20] In another embodiment, there is a network router that stores computer executable instructions that, when executed by the router, perform a set of steps. The network router stores routing information received from a network application. The routing information comprises a message field, a field condition, and a message instruction. The network router receives and processes a network message by comparing the network message to the stored routing information. When the received message's message field meets the field condition, the network router performs the message instruction.

[21] Another embodiment of the invention provides a virtual computer network. The computer network includes a plurality of computers, each configured with at least one transport adapter that converts messages between a transport layer protocol and a network protocol, and a message dispatcher that routes and dispatches messages based on an arbitrary portion of the message's contents. The message dispatcher in each computer routes messages in the virtual network protocol over the transport layer protocol using the transport adapter(s).

[22] In another embodiment of the invention, there is a virtual network that includes at least one virtualized component inserted between layers 6 and 7 of an OSI protocol stack.

BRIEF DESCRIPTION OF THE DRAWINGS

[23] Figure 1 illustrates a block diagram of two nodes in a virtual network according to an embodiment of the invention.

[24] Figure 2 illustrates a block diagram of a virtual network dispatcher according to an embodiment of the invention.

[25] Figure 3 illustrates a composable message protocol according to an embodiment of the invention.

[26] Figure 4 illustrates a block diagram of a virtual network according to an embodiment of the invention.

[27] Figure 5 illustrates a block diagram of a virtual network according to another embodiment of the invention.

[28] Figure 6 illustrates a block diagram of a computer readable medium storing computer software according to an embodiment of the invention.

[29] Figure 7 illustrates a suite of services provided by a virtual network according to an embodiment of the invention.

[30] Figure 8 illustrates a flowchart for performing message delivery according to an embodiment of the invention.

[31] Figure 9 illustrates a modified OSI seven layer network protocol stack according to an embodiment of the invention.

[32] Figure 10 illustrates data flow through a virtual network dispatcher according to an embodiment of the invention.

[33] Figure 11 illustrates network architecture according to an embodiment of the invention.

[34] Figure 12 illustrates a sample virtual mapping table.

[35] Figure 13 illustrates the sample virtual mapping table of Figure 12 after the system has adapted to a machine failure.

DETAILED DESCRIPTION OF THE INVENTION

[36] Message resolution in a virtual network can be accomplished through the use of virtual locations in combination with a universal enabling component, referred to as a virtual network dispatcher (VND), which is included in every resource that participates within the virtual network. A resource may be defined as any message endpoint. With reference to FIG. 1, every device 101-102 on a virtual network 113 is given a virtual address to which its actual address (such as an IP address, MAC address, URL, or other location identifier) may be mapped. The VND 103 may comprise a router module integrated within each device that, using either hardware or software, responds in the same manner to a message regardless of the device on which the router is installed. That is, the router module is device independent. VND 103 includes message handlers 109, further described below.

[37] Message handlers 109 may vary from machine to machine, or they may be consistent across machines. Each message handler may be explicitly configured for a specific type of machine, or may be configured to specifically operate or not operate on any given machine.

Some message handlers may be broadly deployed, e.g., message header handlers and encryption handlers, while other message handlers may only be deployed on a single machine.

[38] Each VND is responsible for performing routing and dispatching functions. Routing is the process of forwarding messages to the device for which they are intended. Dispatching is the process of, upon receiving a message, executing the proper handler (e.g., a software module, function, application program, routine, etc.) or other executable software, in response to receiving the message. The handler that the VND executes may be a routing handler that determines how the VND should process and route the message, or the handler may send the message to an application program running on the device.

[39] In one embodiment, the VND may route and dispatch XML-based messages in an open, extensible messaging protocol that allows distributed, decentralized applications to exchange information regardless of the operating system, object model, or language that each particular application uses. Any protocol may be used that supports the transport(s) used by the virtual network. The VND may be used in conjunction with network transport protocols 107, e.g. TCP, IP, UDP, HTTP, SMTP, SOAP-RP, etc. As messages are received at a location via any transport protocol, the message contents are extracted by a transport adapter 105, and input into VND 103. Each transport adapter receives as input a message formatted according to a predefined transport protocol, and converts (or strips) the message headers to comply with the virtual network protocol. As shown in FIG. 1, each VND 103 may be connected to multiple transport adapters $TA_1 - TA_n$ for use with multiple transport protocols $T_1 - T_n$. This allows each VND to be used across multiple transports, without tying the virtual network to a single transport protocol.

[40] By using multiple transport protocols and protocol adapters, placing a VND 103 on each device provides a platform through which any application program may transparently communicate with another application program independently of the transport layer protocol by using the virtual network protocol. Known in the art are specific application programs that have been configured to communicate over multiple protocols. However, each

application program that does so must be specifically configured. Using the virtual network described herein, applications may communicate over multiple protocols without any special configuration, and without even being aware that communications are being transported over multiple protocols. The VND 103 in each specific device sending each message makes the decision regarding the protocol over which to send the message. For each message, a VND may determine which protocol to use based on one or more predefined protocol policies. Alternatively, the protocol used may be dependant on the application or web service driving the VND for the particular message, based on the needs and/or requests of the application or web service.

[41] When a new transport protocol is developed or needed by the virtual network, a new transport adapter may be created and installed for use with each VND. In this manner, the virtual network may take advantage of new transport protocols, without requiring support for each transport protocol to be built into each application in the virtual network. Instead, support for a new transport protocol is provided to each VND, which contains logic instructing when and how to use the new transport protocol in conjunction with the new transport adapter.

[42] The VND unifies local and remote dispatch on a single machine. Unlike conventional networks where only specified or dedicated machines act as routers, typically every device acts as a router in a virtual network. For instance, a device may receive a message, only to determine that the message should actually be delivered somewhere else in the network. When this occurs, the machine forwards the message to the correct recipient, or to that recipient which the machine believes is the correct based on its present routing tables and rules, instead of (optionally) returning an error message to the message sender.

[43] FIG. 10 illustrates a message routing example according to an embodiment of the invention. A VND 1001 receives incoming message 1003, with FROM field populated with 1.2.3.4, via transport adapter 1005a. VND 1001 may include multiple transport adapters 1005a, 1005b, and 1005c for use with multiple transport protocols. VND 1001 processes received message 1003 using handlers 1007-1013, each of which instructs VND 1003 to

route and/or dispatch messages based on predefined criteria. VND 1001, based on handler 1007, modifies the message's TO field to 7.7.4.4, and outputs routed message 1015 through transport adapter 1005c. Routed message 1015 includes a TO field populated with destination address 7.7.4.4, based on incoming message 1003's FROM field indicating 1.2.3.4.

[44] Because each device acts as a router, a self-healing system may be implemented. When one machine goes down, other machines will automatically compensate and find other paths through which to send messages, making the virtual network fault tolerant. In one embodiment, machines may be placed in redundancy groups. Each machine in the redundancy group can detect that any other machine in the group has failed and left the group. The remaining machines may then update information in one or more message handlers that forward messages to avoid using the machine that is known to be down. Machines can thus compensate for network faults and errors according to instructions encoded in their routing and logic tables, and as further described below. In another embodiment, one or more machine subsystems may be monitoring the network to determine optimal paths and failed paths.

[45] With reference to FIG. 2, handlers 109 contain logic instructing VND 103 how to process messages, i.e., how to handle incoming messages, how to respond to messages, and how to forward messages. For instance, a first handler 109_a may perform virus checking, a second handler 109_b may perform security functions, a third handler 109_c may perform reliability functions, etc. An unlimited number of handlers 109 may be used, as illustrated in FIG. 2 by 109_n. New functionality and capabilities may be added to the virtual network by adding a new handler 109 at any given time, without having to modify network applications on each machine. Processed messages are output through logical recipient ports 111. Logical endpoints may be mapped to any physical port on the device from which the message is being sent.

[46] VND handlers 109 may be created such that, in a virtual network, the application network presents to each application a set of logical endpoints, i.e. devices. Applications

may communicate with endpoints via messages that use non-uniform semantic rules. For example, a first application may send a message over the network to a second application according to a first message format (e.g., headers and data). One or more handlers can modify the message syntax such that the message is modified before it is received by the second application, and appears in a different message syntax. The endpoints may be mapped onto a physical network that may have varying implementations at each endpoint (i.e., different vendors may provide software and hardware to operate the virtual network once the specifications are publicly available), and may communicate using non-uniform transport protocols between endpoints.

[47] Using the above-described network platform, a virtual network may be configured to be self-organizing. That is, the virtual network may be configured to recover from, adapt to, or reorganize itself in response to a specified event on the network. An event could be any predefined condition that triggers the network self-adaptation, including the nonoccurrence of a condition. For instance, the network may be configured to reorganize when it detects that a node of the network has failed. When this event occurs, one or more handlers may instruct the VND to route packets to a new location. In another example, when load on a network path is high, VNDs may route messages over lesser-trafficked network paths. Alternatively, a user may reorganize the virtual network via a graphical user interface, or other configuration interface.

[48] An application programming interface (API) 115 can be provided, through which application programs may interface with the VND 103. Application programs can be written for the computer's execution engine (e.g., an operating system or a virtual machine) that interfaces using API 115 to configure the VND to respond to each message based on the message contents and/or based on the transport protocol on which it was received. The VND mediates the interaction of the protocol and the underlying execution engine.

[49] FIG. 3 illustrates a composable message format used in an embodiment of the invention. Each message 301 includes a header portion 303 and a data portion 305. Headers include information about the information contained in the data portion. That is, headers are

a type of metadata pertaining to the data portion 305 of the message 301. Neither the header portion nor the data portion is required to be a fixed length. The headers may include routing headers 307, reliability headers 309, security headers 311, and application headers 313. Routing headers 307 may include, e.g., a sender's address, a recipient's address, and any other information directed towards routing processes. Address fields may be populated with the virtual address of the entity or device to which the address refers. A virtual address is a logical address to which a physical or other logical address may be mapped. Reliability headers 309 may include any headers that ensure that packets arrive at their intended destination. Security headers 311 may include any headers that ensure that the message contents are not compromised to non-intended recipients. Integrity headers may be included with security or reliability headers, based on a particular application's needs. Application headers 313 may include any headers not previously accounted for, as defined by a network application.

[50] In one embodiment, the message protocol is a composable protocol in that application programs can add new functional aspects as needed without interrupting the processing of preexisting message functionality. In one embodiment, headers are used to provide the new functional aspects. New functional attributes may be stored in one or more message headers. That is, new headers may be added to the existing message without disturbing the processing of the previous message, unlike conventional message protocol suites whereby one message protocol encapsulates another message protocol in order to include a new header (or functional attribute). Thus, the message protocol is extensible in that additional header fields may be added or removed by an application as needed to provide new functionality. This allows network applications to define new header fields and incorporate them into the message format without requiring that every network application be reprogrammed to understand each new message header. Each application program uses only those headers that that specific application program is configured to understand. It may ignore those headers that it does not understand or cannot properly interpret.

[51] The composable protocol may be a modified XML-based protocol, or it may be a modified TCP protocol whereby the additional headers are inserted into the data portion of

each TCP message. When an application adds a new header to a message, the application may send a message to one or more VNDs that instructs each VND to create one or more handlers to route and/or dispatch based on the new header.

[52] Each VND 103 may make routing decisions based on any header and/or data field within each message, or any combination of header and/or data fields within each message. Additional or fewer types of headers may be used. Each handler in each VND 103 provides instructions for routing based on message content.

[53] For example, an application program may define and include an “action” header in each message to indicate the action that a user requests of a recipient. If a network user specifies the action subscribe and sets message data 305 to “baseball scores,” the message may indicate to a first server that the sending user wants to subscribe to a baseball scores email list. Further, the action field may be populated by a virtual function name, mapping to a specific function at each machine on which it is received. If a network user specifies the action subscribe and sets message data 305 to “MSDN,” the message may indicate to a second server that the sending user wants to subscribe to a physical magazine entitled *MSDN Magazine*. Thus, two applications may both use the action subscribe, each in a different manner, as defined by their respective subscribe functions.

[54] In another example, with reference to FIG. 11, suppose an application service provider (ASP) provides three levels of service to customers. The ASP may route messages to one of three different servers and/or applications, based on a level of service to which a customer has subscribed. The ASP may define and use a new application header called `service_level` or the like to indicate a level of service for each customer. The client application may populate the service level field with one of gold, silver, or bronze to indicate the level of service for which the specific customer has paid and/or otherwise subscribed. A master server 1105 may receive all incoming messages from customers 1101 via a network 1103. The master server dispatcher, e.g., VND 103, may then route the incoming customer messages based on the service level. Customers that order gold service may be routed to a fast response server 1107, a server that supports a complete set of services, or other premium-

level server. Customers that order silver service may be routed to a mid-speed response server 1109, a server that supports selected services in addition to basic services, or other medium service-level sever. Customers that order bronze service may be routed to a slow response server 1111, a server that supports only basic services, or other low service-level server.

[55] In another example, an application may include a header field named `geographic_zone` relating to a sending user's geographic location. Routing decisions may then be made based on the sender's physical location, so that messages are sent to a server located closest to the sending user. For instance, in a system that maintains two servers, the device may route a message to a first server in Seattle, Washington when the geographic location header field indicates the sending user is located in Portland, Oregon, and may route the message to a second server in Chicago, Illinois when the geographic location header field indicates the sending user is located in Detroit, Michigan. This avoids sending messages to distant servers when an equivalent server is nearby.

[56] In another example, when a denial of service attack has been launched against one or more machines, a handler may be created that rejects all messages based on a 'credentials' field of each message. The 'credentials' field may or may not be the same as the message's 'from' field. For instance, the 'credentials' field may include a sending user's name, as well as authentication to prove that the sender is who he says he is, whereas the 'from' field may simply include a sender's IP address or logical name. The handler may be configured to accept a message when the message's 'credentials' field contains proper credentials for the sending machine. When a message is rejected, it may be completely ignored, freeing up resources to respond to legitimate messages.

[57] In another example, with reference to FIG. 4, a virtual network may be configured to be self-healing. A machine 401 located behind firewall 403 may receive incoming messages on network connection 404. Machine 401 may be connected via a virtual network (i.e., a physical network configured with adaptive dispatchers and transport adapters as discussed herein) to other machines 407, 409, and 411. Each machine 401, 407, 409, and 411 includes

VND 405. Each machine 401, 407, 409, and 411 may store one or more virtual locations. That is, machine 1 may store and execute a server known as 'alpha.' Machine 2 may house and store a server known as 'bravo.' Machine 3 may house and store servers known as 'charlie,' 'delta,' and 'echo.' Machine 4 may house and store servers known as 'foxtrot' and 'golf.' Each VND in the virtual network is configured with handlers that map each virtual location to its respective physical machine. For example, when device 401 receives a message directed to virtual location bravo, a virtual location mapping handler in device 401's VND instructs the VND to route the message to machine 2. However, because device 401 is the incoming source at a firewall, VND 405 in machine 1 may be configured with additional handlers to first check all incoming messages for viruses and to perform other security measures.

[58] In order to make the virtual network self-healing, handlers may be created to regularly poll another machine or server to determine its network status. That is, where server 'golf' is a backup server for 'echo,' machine 4 may be configured to poll machine 3 at regular intervals in order to confirm that machine 3, and specifically server 'echo,' is functional. When machine 4 does not receive an acknowledgement from machine 3 (or 'echo') within a specified amount of time, e.g., ten seconds, machine 4 may initiate a failover sequence, whereby machine 4 begins sending routing messages to each machine's VND, indicating to each VND that when a message is received for 'echo' on machine 3, the message should instead be sent to 'golf' on machine 4.

[59] Also using the architecture described in FIG. 4, when a server moves from one machine to another, e.g., from one IP address to a second IP address, the virtual mappings may be updated in each VND without requiring each application program operating on the network to be reconfigured. As application programs send messages to the server, each VND automatically reroutes the message to the server's new location. The virtual mappings may be updated manually, e.g., as a result of a new server being added to the system, or the mappings may be updated automatically, e.g., as a result of an automatic healing or adaptation event as described above. FIG. 12 illustrates a sample virtual mapping table. FIG. 13 illustrates the same table after a first machine hosting the www.foo.com website failed and

the system adapted to the failure as described above, rerouting messages to another machine within the redundancy group.

[60] With reference to FIG. 5, the virtual mappings also facilitate easy setup and testing of new servers and network applications. For example, server bravo on machine 2 may be a production email server (i.e., it is an email server presently used in the virtual network). The owner of the virtual network may want to test an upgraded email server with new or different features. Generally, in order to test the new server, a user would have to direct his email client to the new server. This may not be inconvenient for a single user, but it may be a major inconvenience to change every user's server name within a large organization when the new server goes live. Using the inventive system, the test server may be installed on machine 3, also named bravo, and referred to as bravo'. Each VND may be configured with a handler that instructs it to route messages for bravo to machine 2. However, the same or a different handler is configured to route messages for bravo to machine 3 when the sender is a predetermined user, e.g., the network administrator that is testing the new server (bravo'). Thus, no reconfiguration of the test user's machine is required. In addition, when the new server bravo' is ready to be put into production, the network mappings may be changed by directing all bravo messages to machine 3, without interrupting any users' email service. Each user will transparently begin using the new email server because the virtual mapping has changed.

[61] With reference to FIG. 7, a suite of virtual network services may be provided to ensure that communications and services in the virtual network are secure, adaptable, reliable, self-healing, and platform independent. Virtual network synchronization services 703 ensure that distributed data within the network remains synchronized. Virtual network eventing services 705 create new routing and/or dispatch rules based on the occurrence or non-occurrence of an event. Virtual network messaging services 707 route messages according to virtual names and locations. Virtual network name services 709 provide name resolution services based on any substring of a virtual name. Virtual network group services 711 manage name-mapping tables. Virtual network addressing services 713 perform naming and routing services for fixed-length address names, e.g., IPv6 addresses. However, virtual

network addressing services may be used with any flat address space of fixed-size addresses. Virtual network routing services 715 route and dispatch based on dynamic rules in conjunction with dynamic headers using a composable message protocol. Virtual network security services 717 may be provided across all levels of the network to ensure that message contents are secure and authentic. Virtual network management 719 may be performed across all levels, such as managing names, routing/dispatch handlers, eventing, etc. Virtual network monitoring services 721 allow a network administrator to monitor network usage, bandwidth, bottleneck points, and the like, as is presently known in the art.

[62] An embodiment of the invention may be based on a modified version of the seven-level open systems interconnection (OSI) network model, as illustrated in FIG. 9. One protocol stack that may be used with the OSI model is the TCP/IP protocol stack. The invention may insert an additional level of abstraction in the OSI network model, or any other network model, by inserting a layer between the top application layer and the layer immediately below the top application layer. The new layer, referred to as the virtual network (VN) layer, should be consistent across all applications so that the applications can interoperate in a uniform way as defined by the VN layer. A network into which a VN layer has been integrated is referred to as a virtual network. In one embodiment, the VN layer includes a virtual network dispatcher and any necessary transport adapters, routing and dispatching messages based on message handlers and a virtual address mapping table.

[63] Using the above-described architecture, a network may route and dispatch messages based on different message content, on an individual message basis. The invention provides a network protocol that programmers may adapt and configure as needed using the API. Programmers, and programs using the API, may instruct VNDs how to route and dispatch incoming messages. That is, programmers send meta-messages to VNDs, where each meta-message is constructed according to the API and provides one or more routing and/or dispatching instructions.

[64] FIG. 8 illustrates a flowchart of a general routing procedure according to an embodiment of the invention. In step 801, a user decides to send a message to a service

known as 'foo.' The machine creates the message to service 'foo' in step 803. In step 805, the service name 'foo' is mapped to a virtual address based on a virtual address mapping table. The message is secured as necessary in step 807. In one embodiment, security is performed using SOAP extensions such as those defined by the Web Services Security Language (WS-Security) and/or the Web Services License Language (WS-License). In another embodiment, a transformation is performed on the message to select relevant parts. A digest is computed over the selected parts and encrypted/signed by the sender. Portions of the message might be confidential. In this case they are encrypted using a shared key or a new key which is, in turn, encrypted for the recipient. In step 809, the virtual address is mapped to a group address (GADDR), when applicable. In step 811, the adaptive dispatcher (i.e., a VND) determines the best target and, in step 813, maps the GADDR to a virtual address. In step 815, the virtual address is mapped to a physical address, and, in step 817, the message is sent to the physical address. The recipient machine receives the message in step 819, and validates the security in step 821.

[65] The inventive methods may be embodied as computer readable instructions stored on a computer readable medium such as a floppy disk, CD-ROM, removable storage device, hard disk, system memory, or other data storage medium. Alternatively, the inventive methods may be embodied in a combination of hardware and software, or in only hardware. Fig. 6 illustrates a block diagram of a computer readable medium 601 that may be used in accordance with one or more of the above-described embodiments. The computer readable medium 601 stores computer executable components, or software modules, 603-613. More or fewer software modules may alternatively be used. Each component may be an executable program, a data link library, a configuration file, a database, a graphical image, a binary data file, a text data file, an object file, a source code file, or the like. When one or more computer processors execute one or more of the software modules, the software modules interact to cause one or more computer systems to perform according to the teachings of the present invention.

[66] While the invention has been described with respect to specific examples including presently preferred modes of carrying out the invention, those skilled in the art will

appreciate that there are numerous variations and permutations of the above described systems and techniques that fall within the spirit and scope of the invention as set forth in the appended claims.

1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434 3435 3436 3437 3438 3439